Assessing the Quality of a Business Process Implemented across Systems of Systems

Carol Woody, Ph. D.
Robert Ellison, Ph.D.

Software Engineering Institute

SSTC

June 21, 2007

The Desire

Real-time shared situational awareness: operational and tactical levels

Decision superiority enabling more agile and survivable joint operations

Concept of operations for an Aerospace Operations Center: “seamless linkage of superior and subordinate elements with the Theater Air Control System, joint force, and external agencies to optimize personnel, functional, and support system capabilities.”

Reduction in costs with the use of shared services

Mission Threads: Work Process Realities

Effective execution of mission threads require integrating system and people activities across a *constantly evolving* mix of changing systems and people

Increased reliance on shared technology/services requires establishing *operational trust* among systems, software components, and services.

Establishing and maintaining support for mission threads requires *traceability* between technical decisions and mission requirements

Reliability of mission threads can be affected by the *interactions* of software systems, hardware systems, and human operations

Mission threads may be adjusted ad hoc to meet immediate critical needs. This level of *flexibility* contributes to their *fragility*.

DoD Mission Thread Characteristics

Can be decomposed into multiple steps that cross technology, people, and organizations

Can be threatened by events that cause disruptions (intentional and accidental) – operational dependability (survivability) is critical

Are highly context sensitive

- Mission thread instantiation is dynamically determined by available resources
- Limited resources may require revisions in mission thread: use of security mechanism may be overridden by critical need for information
- Systems may to be configured to support a special instantiation of a mission thread.

Scale: Operational System Challenges

Characteristics of large networked systems

- Heterogeneous, potentially inconsistent, and changing elements (hardware, software, systems)
- Continuous evolution of functionality and usage (perpetual beta)
- Erosion of the people/system boundary (each influences the other)
- Independently developed and managed systems integrated into a system of systems

Mitigating component failure is not sufficient

- Increasingly failures result from a group of errors (operator, unexpected software state, and user) that can be addressed individually but not collectively
- Increasing dependencies among development, deployment, and operations (complexity hides risks until deployment)

Complexity is often addressed with segregation and simplifying assumptions

- Hides risks making them difficult to observe until deployment.
- Conflicting assumptions lead to mismatches

Systems of Systems

Mission Threads depend on the ability to integrate multiple systems

- Effects of individual system failures on mission thread
- Systems developed at different times with variances in technology and expected usage
- Systems are not be constructed from uniform parts: particularly as systems are extended, patched for security flaws, and repaired to address other errors.

Normal usage and attack methods change more rapidly than systems

- New usage implemented by using existing resources rather than developing new ones – shared services.
- Systems evolve rather than being replaced – legacy is more the norm

Failure Management

Consider errors as a *normal* event.

- Inconsistencies (mismatches) must be assumed with the integration of independently developed and administrated systems.
- Changes in usage may lead to unexpected system or system of systems behavior or to a new error state for a specific system.
- Erosion of the people/system boundary – people are part of the system. Operator and user behavior affect systems behavior and may lead to system failures.
- For software systems, failures may be caused by a combination of relatively minor errors rather than by a component failure.

Complex Failure: 2003 Power Blackout [1]

On August 14, 2003, approximately 50 million electricity consumers in Canada and the northeastern U.S. were subject to a cascading blackout. There was not a single cause for this event.

The blackout was initiated when three high-voltage transmission lines went out of service when they came into contact with trees too close to the lines.

The loss of the three lines resulted in too much electricity flowing onto other nearby lines, which caused those lines to overload and then be automatically shut down.

Independent monitoring system was not set up to consider the effects of an out-of-service line and had to be manually adjusted.

Complex Failure: 2003 Power Blackout [2]

The failures occurred over a four hour period

- Tree trimming procedures were not followed
- Race condition disabled alarm system that provided the only effective means for grid operators to identify problems. The corruption of the data stream caused the backup server to fail also.
- Alarm subsystem could only be restarted by restarted full control system – sixty minutes. Without the aid of the alarms, grid operators were not aware of affects of the loss of the lines.
- IT confused by initial symptoms. Did not notify grid operations of the alarm subsystem failure.

Complex Failure: 2003 Power Blackout [3]

The power failure demonstrates the need for a system assurance case to include not only the computing systems but also business operations, training, and IT operations.

- Issues with operator training managing emergency conditions.
- Operational and system analysis should have identified a system requirement to automatically notify grid controllers when the alarm system or other critical subsystems fail.
- An analysis of software faults in addition to hardware faults might have lead to a business continuity requirement to be able to restart a service such as alarm notification without having to restart the entire system.

The events leadings to the blackout all had non-malicious intent, but could have been exploited especially with some insider knowledge.

Importance of System Quality Attributes

For networked systems, the initial effort often concentrates on meeting the functional requirements under normal usage – sunny-day scenarios.

But the military operational environment can stress software systems significantly more than the typical business setting.

- integrate system and people activities across a *constantly evolving* mix of changing systems and people
- establish *operational trust* among systems, software components, and services – predictable execution of a mission thread
- enable ad hoc changes in mission thread and available resources to meeting an immediate critical need

An Approach for Quality Analysis

Context and change are critical to quality analysis

- Select the qualities to be evaluated with respect to operational change
- Selected qualities must be clearly defined for the context
- Build one or more detail operational work process flows incorporating information about the selected qualities

Complexity is unavoidable but analysis cannot consider everything

- Success of the operational work process flow must be defined
- Focus on the areas most critical to operational success

Failure is assumed to result from a combination of small problems that drive operational execution outside of expected behaviors

- Failure potentials can be characterized as stresses
- Stresses exhibit behaviors that can be identified and monitored
- Mitigation is focused on stress management

Traceability: Translating Operational Qualities into System Qualities

Operational qualities for business case

- Operational continuity
- Regulatory compliance
- Maintainability (as usage changes)
- Time-to-deployment for supported functionality
- Usability / Performance (these are visible to operations)

Associated System qualities: software engineering perspective

- Reliability
- Flexibility / Expandability (Extensibility)
- Security - Availability
- Interoperability
- Verifiability (software and information assurance for regulatory compliance)

Survivability Analysis Framework (SAF) -1

Framework for survivability

Focus on Usage – a work process being executed in a well-defined context

SAF Process:

- Identify a work process (mission thread)-specific instantiations
- Define successful completion criteria for the process
- Describe critical steps required to complete the process (end to end) - sequenced activities, participants, and resources
- Describe how the work process can be compromised at each critical step and with the composition of activities
- Analysis of overall process thread to see how responses at each step may affect the overall success of the thread

Survivability Analysis Framework (SAF) -2

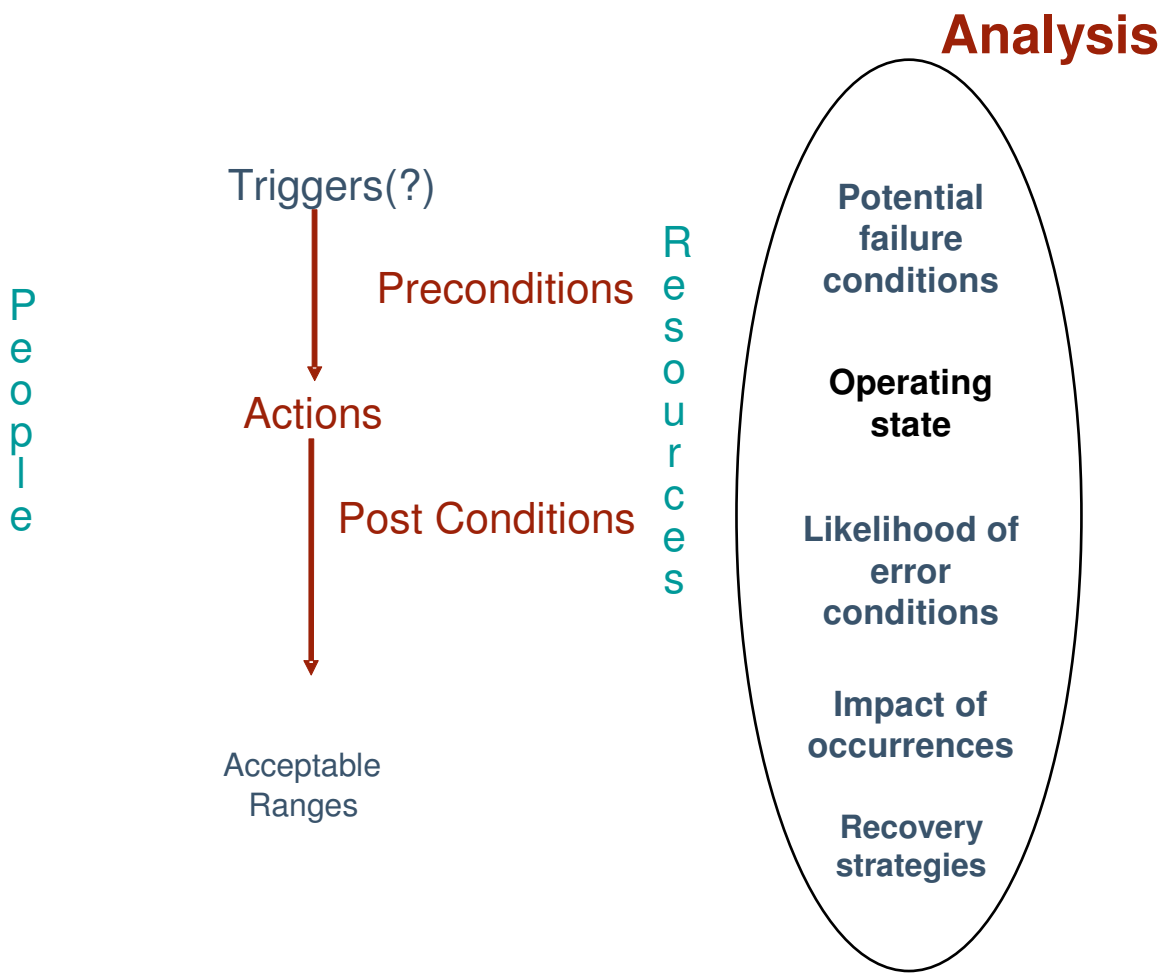
Success goals are affected by stresses that individually or collectively push the operational thread beyond the limits of acceptable degradation and recovery.

Given the potential complexity of a work process thread, identification and analysis of all possible stresses is NOT feasible. Analysis of a critical sample will be considered a sufficient indicator.

The following stresses linked to survivability were characterized:

- Interaction (data)
- Resources
- People

Mission Step Analysis



Time-sensitive Targeting Scenario

Army unit on patrol spots a missile launcher preparing to fire.

Commander is notified and report is sent to JFLCC where it is shared with other Intel points and designated as TST target.

F16 recommended by JFACC is selected to strike the target.

Air Force completes the engagement and reports results to other Intel points.

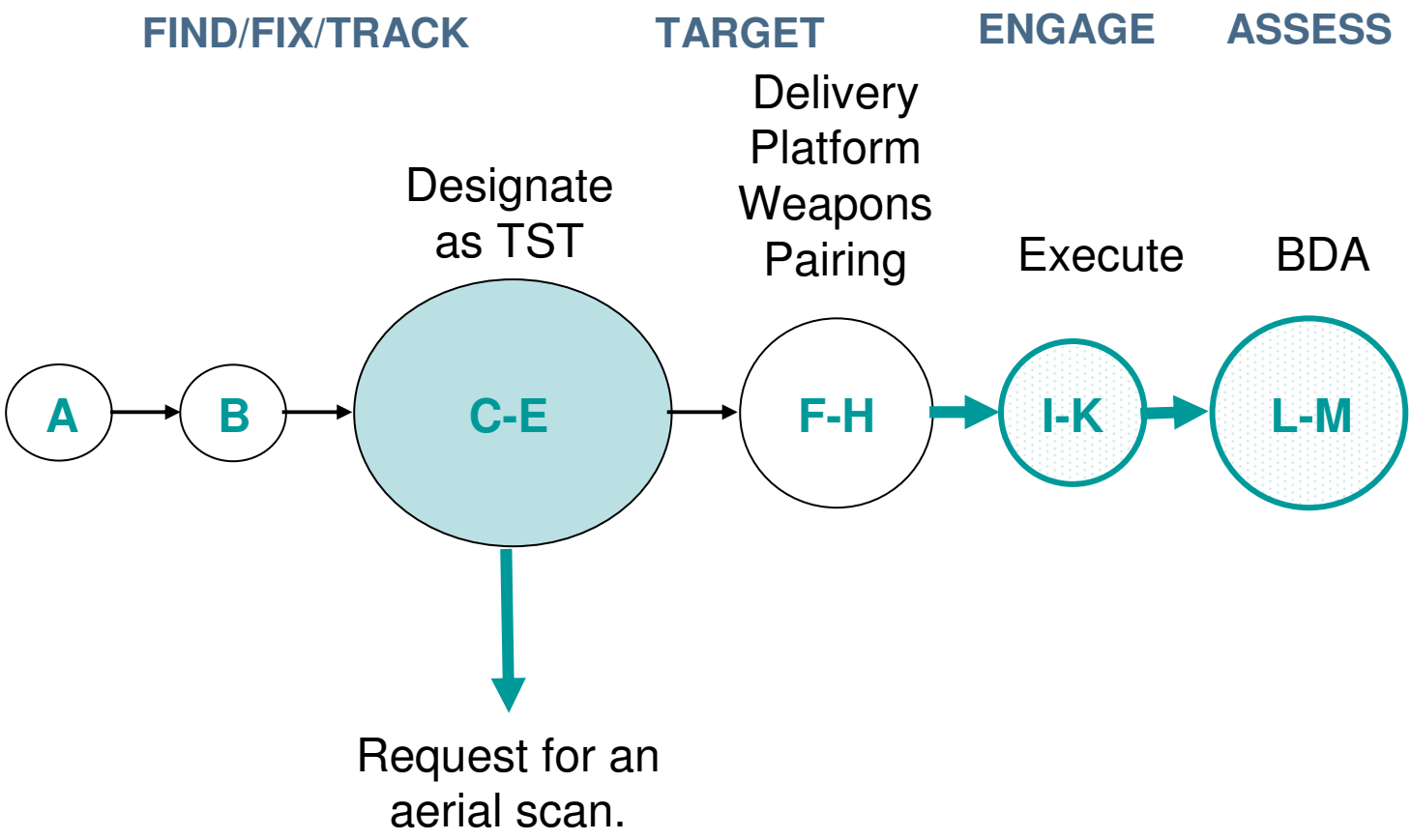
Initial Steps in TST Scenario

A: An army unit on patrol spots a potential TST (for the scenario, they see a missile launcher preparing to fire).

B: The unit contacts their command post and provides a report. Regardless of whether the report is verbal or digital, the report is recorded digitally and, based on the commander's guidance, is forwarded to the JFLCC TST cell.

C: Through the JADOCS TDN tool, the JFLCC TST cell shares information with other TST cells. TDN is used to collect information from other intelligence sources. Other resources may be diverted from existing missions to provide further intelligence on the target. Data collection requirements are specified.

TST Mission Thread



Time-line of Actions - People

Describe how each person is involved in an action

Action / People	A: Sighting	B: Report Sighing	C: Sighting Analysis	D: Decide target is TST
Army Unit	Sight Target	Report potential TST to Command		
Army Command Post		Based on guidance report to TST Cell		
JFLCC TST Cell		Enter potential TST into TDN	Use TDN to collection additional information on target.	Confirm the target as a TST. Enter into JTSTM

Resources

Describe the resources involved in an action

STEP	Resources
A: Sighting	Physical location and mission orders, sight of the missile launcher
B: Report Sighting	Transmitted info (e.g., who, what, where) by some means (e.g., tactical radio), potentially access to JADOCS terminal or some other mechanism to contact JFLCC TST cell Commander's guidance matrix determining what is and is not a TST
C: TST Analysis	Access to JADOCS terminal to use TDN Other assets Inputs from other TST cells Control of surveillance assets to provide more data about target

Step C: TST Cell Analysis - 1

Preconditions	Presence of an object of interest Nomination as a TST Knowledge of Joint Forces Commander's intent

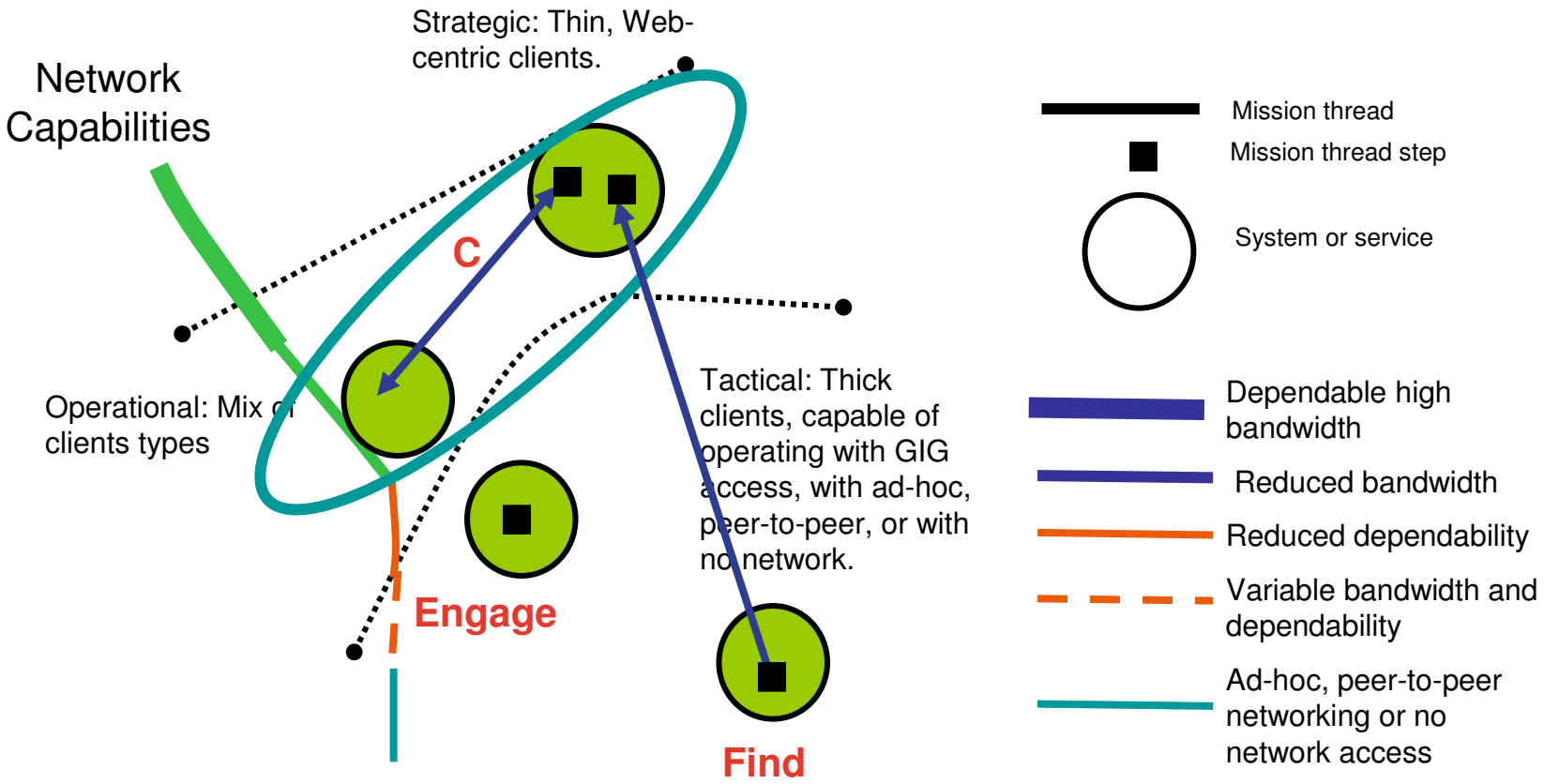
Step C: TST Cell Analysis - 2

Actions	All TST cells: •Sharing, interaction, and preliminary planning with other intelligence staff •Further actions (e.g., data collection) are identified Data is gathered continually until sufficient data exists to make a decision concerning the potential target. This may require use of: •additional surveillance assets •further verification actions •requests for assessment of target value.

Step C: TST Cell Analysis - 3

Post Conditions	Preliminary collateral damage assessment Timeliness of target (deadline for action)

TST Mission Thread – Network Constraints



Potential Survivability Analysis Outcomes

From initial use of the framework:

- Potential points of failure (stress analysis)
- Survivability gaps (step interactions)
- Mitigation strategies for a work process
- Gaps in current component requirements
- Better quality specifications for component requirements
- Better quality specifications for shared services

Application of the framework to a work process thread periodically as systems and services change:

- Changes in survivability capabilities over time
- Opportunities for survivability improvement

Lessons Learned so Far

Exposing developers to the operational realities increases consideration of those issues during design and implementation.

Operational personnel view this as an effective means of communicating their challenges to management

Characterizing a work process through the interactions of people, resources, and activities provides a structured way of describing the complexity

Identification of potential failures requires detail knowledge of how activities are actually performed

Analysis steps are unstructured – limited repeatability

Future Use

Change management evaluation: consider the impact of a change to a work process

Establish an approach for the construction of assurance cases and identification of evidence that assurance is provided